Machine Learning for Robust Control and Decision Problems

Subramanian Ramamoorthy
School of Informatics
The University of Edinburgh

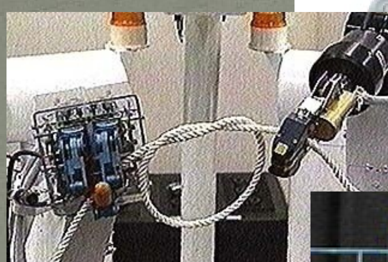
29 October 2008

Motivation: Robust Autonomy

2003

2008

[Havoutis – PhD, Clossick – MSc]



[Komura, Larkworthy - PhD]

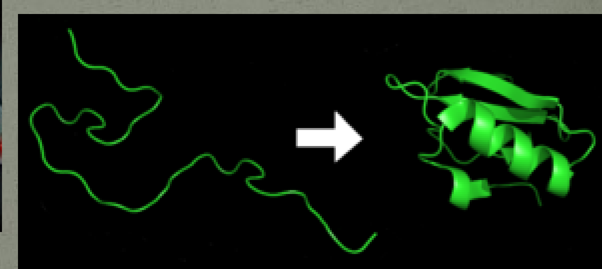
[Prabhakaran - MSc]

2010
& beyond

Autonomous Robotics

[Savani @ Warwick, Elder - MSc]

Autonomous Trading Agents



Computational Biology

State of the Art: ML in Control & Decision

- Supervised Learning
  - Regression - mapping between states and actions (everything from neural networks to Gaussian processes), forward/inverse models
  - Dynamic time series modelling - HMM, SLDS, etc.
- Un-supervised Learning
  - ‘Feature’ extraction, e.g., animators need a few key variables/knobs
- Reinforcement Learning
  - Planning and control is the *raison d'être*

Open issues w.r.t. goal of robust autonomy:

Hard to acquire *dynamically dexterous* behaviours over large domains

- Even when ML sub-component seems rigorous, there may be little understanding or leverage over global task-level dynamical properties
- *What does Asimo need to know, so he can compensate for a hard push*?

Lattice of Control and Decision Problems

Regression, Reinforcement learning

(X,U,W)
Robust control
Game, adversary, strategy

Real-world apps need robust control (arbitrary W, high-dim, constraints) – but it isn't well studied in ML context

(X,U)
Feedback control & optimality

(X,W)
Verification

(X)
Motion synthesis & planning

Time series models, PCA/NLDR

Robust Control and Decision Problems

- Robust control ⇔ Compute *Best Response* in a *differential game against nature* or other agents (especially when application requires interaction and autonomy)

Dynamics	$\dot{x} = f(x,u,\mathbf{w})$
Performance	$J(x,u,\mathbf{w}) = q(x_T) + \int_0^T g(x,u,\mathbf{w})dt$
HJI Equation	$sol.\{\nabla_t V + \nabla_x V \cdot f(x,u,\mathbf{w}) + g(x,u,\mathbf{w})\} = 0$

Solution concepts : viscosity, $\min_u \max_w \equiv \max_w \min_u$, viability, etc.

  - **w** are *structured large deviations* (beyond sensor noise…)
  - Constrained high-dim partially-observed problem is hard!
    - How can we extend ML methods to address this?

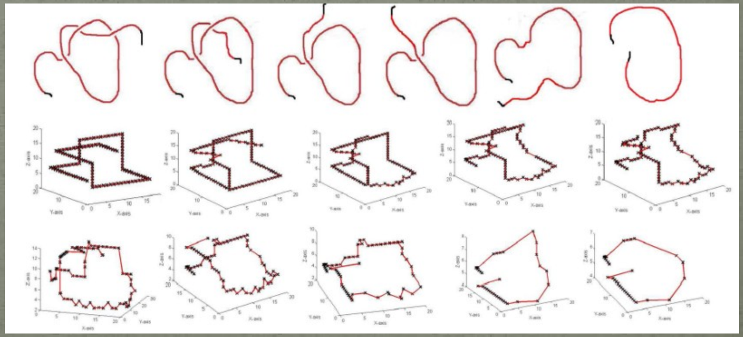
Making Robust Control Tractable

- *Multi-scale Problem Formulation:*
  1. *Sufficient Abstraction* – dynamical primitives that suffice to answer qualitative questions, e.g., reachability
  2. Use abstract plan to constrain on-line solution
- *Data-driven Solution*:

  Learn from on-line exploration and/or demonstration

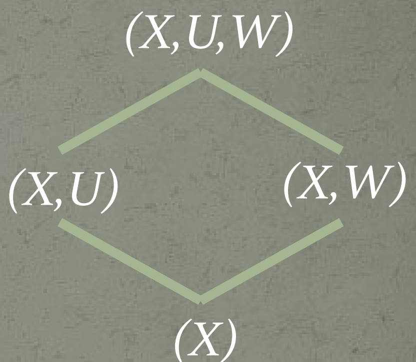
Example [Prabhakaran, MSc '08]:

- To unknot a rope, first decide using topology (level sets of knot energy)
- This constrains space for motion planning and reinforcement learning
  - *10* x faster computation



Research Problem 1: How to Abstract Dynamics?

- Nonlinear dimensionality reduction & latent variable methods focus on metric properties at bottom of lattice - *X*
- How can reductions preserve other dynamical properties?
  - Behavioural (I/O dynamics) equivalence
  - Similar reachable sets given disturbances



Related Prior Work:

- Pappas & Tabuada: Bisimulation
- Amari & Ohara: Differential geometry of systems
- Recent work on use of symmetries & homomorphisms in RL

Research Problem 2:
How to Acquire Large-Scale Strategies?

- Switched dynamical systems are obvious candidates
  - However, we also need *autonomous transitions* & *concurrence*
- Architecture should support meaningful inferences about *global dynamical behaviour*
- Can we do active learning to refine such models?
  - splitting, merging, simplification using abstractions

Related Prior Work:

- Extensive literature on (flavours of) HMM, SLDS, etc.

  Also, cPHA work by Tomlin, Williams, et al.

Research Problem 3: How to Constrain On-line Control?

- Related to shaping in Reinforcement Learning
  – but we want more leverage over dynamical behaviour
- Approaches in the tradition of optimal control theory:
  - Data-driven multi-scale solution of the HJI equation
  - Alternate game-theoretic solution concepts

Related Prior Work:

- *Adversarial* RL via stochastic games
  - Littman, Uther & Veloso, Filar & Vrieze, etc.

Some Questions for Discussion

In the spirit of recent work on robust control, the exercises in our earlier paper analyzed the performance of policy rules in worst-case scenarios, rather than on average. However, the more conventional approach to policy evaluation is to assess the expected loss for alternative policy rules with respect to the entire probability distribution of economic shocks, not just the most unfavorable outcomes.

[B. Bernanke, M. Gertler, Should central banks respond to movements in asset prices? *American Economic Review*, To Appear]

- *How to get "entire probability distribution of shocks"?*
- *Why model (prediction vs. qualitative behaviour)?*
- *How might a learning algorithm incorporate policy concerns?*